



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Logvinov et al.

U.S. Serial No.: 10/697,629

Filed: October 29, 2003

For: HIGHLY PROGRAMMABLE
MAC ARCHITECTURE FOR
HANDLING PROTOCOLS THAT
REQUIRE PRECISION TIMING
AND DEMAND VERY SHORT
RESPONSE TIMES

Group Art Unit: 2661

Examiner:

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Mailing Label No EV522733151

Date of Deposit: September 9, 2004

I hereby certify that this paper or fee is being
deposited with the United States Postal Service
Express Mail Post Office to Addressee service under
37 CFR 1.10 on the date indicated above and is
addressed to the Commissioner for Patents,
Alexandria, VA 22313-1450.

Print Name: Lori Goldman

Signature: *Lori Goldman*

Mail Stop Missing Parts
Commissioner for Patents
Alexandria, VA 22313-1450

PETITION UNDER 37 C.F.R. § 1.47(a)

This Petition is respectfully submitted under 37 C.F.R. § 1.47(a) to request that the above-identified application be made by joint inventor, Mr. Oleg Logvinov, on behalf of himself and joint inventor, Mr. Fred Skalka.

Mr. Skalka, after diligent efforts were made on behalf of Mr. Logvinov, refused to sign the Declaration and Power of Attorney ("Declaration/POA") that must be filed in response to the Notice to File Missing Parts of Nonprovisional Application Filed under

37 CFR 1.53(b) – Filing Date Granted (“Notice”) dated February 9, 2004 that was received in the above-identified application. Accordingly, the joint inventor Mr. Logvinov desires to prosecute the above-identified application on behalf of himself and Mr. Skalka.

The above-identified application claims priority from U.S. Provisional Application No. 60/422,026 filed October 29, 2002. Attached as Exhibit A to this Petition is a copy of the application as filed on October 29, 2003, to which is attached a copy of the Declaration/POA executed by Mr. Logvinov¹. The Declaration/POA identifies Mr. Skalka as a joint inventor, with his signature block left blank. Accordingly, the Declaration/POA is executed by all available joint inventors on behalf of the nonsigning inventor. See M.P.E.P. § 409.03(a).

Attached as Exhibit B to this Petition is the Declaration of Mr. James Reeber (“Reeber Decl.”). Mr. Reeber is the Director of Marketing of the current assignee, Arkados, Inc. (“Arkados”), of the above-identified application. The Reeber Decl. sets forth the efforts Mr. Reeber took on behalf of Mr. Logvinov and the current assignee to have Mr. Skalka execute the Declaration/POA.

At the current time and at the times when Mr. Reeber made efforts to have Mr. Skalka execute the Declaration/POA, Mr. Skalka has been involved in a dispute with Enikia, LLC (“Enikia”) concerning his employment with Enikia. Reeber Decl. ¶ 4. The

¹ The original Declaration/POA executed by Mr. Logvinov is submitted herewith in response to the Notice.

above-identified application, previously owned by Enikia, is now owned by Arkados.

Reeber Decl. ¶ 1.

On July 26, 2004, Mr. Reeber sent to Mr. Skalka, via email, an original Declaration/POA for the above-identified application identifying the two joint inventors and requesting that the Declaration/POA be executed by Mr. Skalka. Reeber Decl. ¶ 9. Mr. Reeber also offered to pick up the executed Declaration from Mr. Skalka. Reeber Decl. ¶ 9. Mr. Skalka's last known address is 630A Palmer Lane, Yardley, Pennsylvania, and last known email address is fskalka@juno.com. Reeber Decl. ¶¶ 5-6.

Mr. Skalka clearly understood that he was being requested to sign a Declaration/POA for the above-identified application. Mr. Reeber, in response Mr. Skalka's statement that Mr. Skalka had "executed" an assignment earlier in the year (Reeber Decl. ¶ 10), in an email dated August 3, 2004 indicated to Mr. Skalka that the documents previously signed were for the provisional application from which the above-identified application claims priority. Reeber Decl. ¶ 12. Mr. Reeber explained in that email that the Declaration/POA that needed to be executed concerns the above-identified application, which was based on the provisional application. Reeber Decl. ¶ 12. Mr. Skalka previously had executed an assignment for such provisional application, which subsequently was recorded with the USPTO. Reeber Decl. ¶ 3.

Following another email inquiry from Mr. Reeber on August 9, 2004 as to whether the Declaration/POA has been executed (Reeber Decl. ¶ 13), Mr. Skalka first responded that he's "waiting to here [sic.] from my lawyer on this" (Reeber Decl. ¶ 14;

see also Reeber Decl. ¶¶ 7-8) and then stated that his lawyer said not to sign anything until we settle. Reeber Decl. ¶ 15.


Mr. Reeber did not subsequently receive an executed Declaration/POA (Reeber Decl. ¶16) and, on September 8, 2004, Mr. Reeber contacted Mr. Skalka by telephone where Mr. Skalka stated to Mr. Reeber that, continuing under advice of counsel, he refused to sign the Declaration/POA for the above-identified application until the litigation with Enikia settles. Reeber Decl. ¶ 17.

Thus, diligent and bona fide efforts have been made to have Mr. Skalka execute the Declaration and Power of Attorney and join in the above-identified application. In view of the correspondence between Mr. Reeber and Mr. Skalka, Mr. Skalka's clear understanding of what has been requested of him and Mr. Skalka's acknowledgement that he refuses to execute the Declaration/POA for the application under advice of counsel, it is clearly evident that Mr. Skalka refuses to join in the application.

Accordingly, the joint-inventor, Mr. Logvinov, respectfully requests that the above-identified application be made on behalf of himself and the nonsigning joint inventor, Mr. Skalka.

Kindly telephone the undersigned at (908) 722-0700 if it is believed that further discussions would help advance the prosecution of this application.

Respectfully submitted,



Davy E. Zonerach
Registration No. 37,267

September 9, 2004
103880-037US

Norris, McLaughlin & Marcus
721 Route 202-206
Somerville, New Jersey 08876
(908) 722-0700

**PROGRAMMABLE MAC ARCHITECTURE FOR COMMUNICATIONS SYSTEMS
REQUIRING PRECISION TIMING AND FAST RESPONSE TIMES**



CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/422,026 filed October 29, 2002, which is incorporated by reference herein.

FIELD OF THE INVENTION

[0002] This invention relates to the field of high speed data communications, and more particularly, to data processing operations performed at the MAC/PHY layer of a data communication system.

BACKGROUND

[0003] Powerline communications (PLC) was selected as an exemplary technology that will be used for illustrative purpose only and it is important to realize that any data communications technology could take advantage of this invention. The use of PLC technology is very attractive because there is no need to install new wires to communicate between stations. Existing power wiring in homes and business as well as the wires used to carry power in the electric power distribution grid are all capable of supporting high-speed data communications. In addition to in-home and access, another key application segment for PLC is multiple dwelling units (MDU) or multiple tenant units (MTU) such as apartment buildings, hotels and motels.

[0004] Each of these different application areas represents a different set of design parameters, but all use a MAC/PHY layer in their transceivers. Each of these different areas is in a different standardization condition and government regulatory stage. In-

home PLC standardization, for one example, is well along with the formation of an industrial alliance, such as the HomePlug Alliance, and the subsequent release of their formal PLC specification. Other segments of PLC applications, such as access, are just starting to become established^[3] (see ETSI TS 101 867 V1.1.1 (2000-11); Technical Specification -- Powerline Telecommunications (PLT); "*Coexistence of Access and In-House Powerline Systems*"; Reference: DTS/PLT-00004; November 2000) and so the specifications are more fluid.

[0005] Using a flexible and programmable architecture for the design of the MAC/PHY layer in transceivers for each of these applications segments would be highly desirable. The programmable flexibility would mean that changes in standards, regulatory requirements, product patches, new product features and product enhancements could mostly be accommodated by installing new software instead of with costly and time consuming hardware modifications (e.g., revising ASIC logic usually means manufacturing a new very expensive foundry mask set for the device).

SUMMARY

[0006] This invention provides a MAC/PHY layer controller (heretofore referred to as the *HardMAC*) that interfaces between a general-purpose processor and hardwired DSP logic. The HardMAC performs tasks whose functions are well defined and are, generally too fast for the processor to perform. The HardMAC controls the hardwired DSP logic in such a way as to simplify and generalize the operation of the logic.

[0007] In accordance with one embodiment of the present invention, a communications transceiver includes a programmable MAC/PHY layer controller (HardMAC) module coupled to a microprocessor and DSP hardware. The HardMAC

preferably is a programmable coprocessor module including pre-defined operation hardware blocks having parameterized functions whose parameter values are programmable. In a preferred embodiment, a portion of the coprocessor module controls timing and the clock cycle rate is a programmable parameter. The programmability of the HardMAC avoids the necessity to make hardware changes involving pre-defined operations performed at a communications transceiver whose parameters may vary based on changes on regulatory requirements or the like.

Thus, a MAC/PHY layer controller is constructed out of three types of blocks: highly flexible general-purpose processor software, very flexible parameterized coprocessor and hardwired DSP logic. The composite PHY function is composed of part of the HardMAC controller and hardwired DSP logic. The composite MAC is composed of general-purpose processor code and a part of the HardMAC. This level of application specific flexibility accommodates a wide variety of alterations including changes to meet new regulatory requirements, solutions to eliminate errors in the operation of the system, and updates for end-product enhancements.

BRIEF DESCRIPTION OF THE FIGURES

Figure 1 – shows an example of how the device might fit in an overall system;

Figure 2 – shows an example of the primary internal blocks for a device;

Figure 3 – is an example of a detailed block diagram of HardMAC internal interconnections; and

Figure 4 – is an example of a detailed block diagram of a command sequencer module (#225).

DETAILED DESCRIPTION

It is noted here that PLC technology is used in this disclosure to help illustrate details of the invention and is by no means the only technology that the invention can be applied to, but can generally be used with any communications technology. A system level view of data communications systems components is shown in Figure 1. The HardMAC (#130) provides a flexible interface between software (heretofore referred to as the SoftMAC) running on the general-purpose processor (#100) and the hardwired DSP logic (#140 and #145) to create a complete MAC and PHY function. The highly programmable nature of the processor and the flexible nature of the HardMAC combine to create a MAC/PHY layer that is flexible and can be adapted for various needs without restructuring the system.

There are nine (9) major blocks within the HardMAC as shown in Figure 2. The System bus (#205) interfaces to the processor (#200) while the hardware PHY logic (#250) interfaces to the six (6) blocks (#210, #215, #220, #225, #230, and #235) as shown. The detailed interconnections between blocks are shown in Figure 3.

1. System bus interface and DMA (#205)

The System Bus Interface and DMA Controller provide a system bus Master Interface with a two Channel DMA Controller and a system bus Slave Interface to all registers in the HardMAC. The DMA controller provides one channel for data transfers to Tx Data FIFO and one channel for data transfers from the Rx Data FIFO. The system bus Slave Interface provides address decode and read data select for HardMAC modules which have register interface and implements all logic to generate

the proper response to a system bus data transfer. The slave is not split transaction capable.

2. TX PHY Data FIFO (#210)

The Tx Data FIFO provides a buffer between the system bus and the Tx PHY. This allows a block of data to be transferred to the Tx PHY and cross the system bus/Rx PHY clock boundary.

3. RX PHY Data FIFO (#215)

The Rx Data FIFO provides a storage buffer for a PLT payload. The Rx Data FIFO also crosses the clock boundary between the Rx PHY and the system bus. It packs the eight bit data from the Rx Phy into 32 bit words that are written into Rx PHY FIFO buffer. It also generates a StrtDmaHdr signal when the header has been received. It also does the DA compare and generates SA and SA ready signals to the DCB CAM.

4. FCS Checker (#220)

The FCS (Frame Check Sequence) Checker calculates the 16-bit CRC of the complete incoming payload section of a received frame using a specific polynomial. A signal is generated that indicates if the CRC check was good or bad. This signal is sent to the MAC/PHY Status and Interrupt controller for use as part of the MAC/PHY status and the possible generation of an interrupt.

The last two, eight bit words written to the MAC by the Rx PHY are available in the FCS register. At the end of the payload receive, these two words contain the FCS of the current receive payload.

5. PHY Command Sequencer (#225)

The PHY Command Sequencer controls the timing and issuing of commands to the PHY from the MAC. This block is software programmable and flexible in how it operates.

The internals of this block are shown in Figure 4 with two elements, namely *Command and Control* (#400) and *Branch and Sequence Controls* (#405).

The PHY Command Sequencer issues a command to the PHY to put the PHY in one of seven states. The commands are set for some time before a timing pulse, called the PHY Sequence Pulse (PSP) is issued to cause the PHY to execute the command at a specified time. The commands and the PSP are issued by the processor by writing to registers or by the sequencer. The sequencer contains 32 entries in a table that is accessed by the Branch & Sequence Controls. These registers contain command information to the PHY and command and control information for the sequencer.

The PHY Command Sequencer consists of two basic blocks, Command & Control and Branch & Sequence Controls. The Command & Control section contains all the logic required to issue the commands and generate the PSP while the Branch & Sequence Controls contains the logic for the sequence controls and the bus interface.

The PHY Command Sequencer runs with different PHYClk rates depending on application needs and this is accomplished with synchronizer blocks.

The Branch Sequence Registers (BSeqR) contain information that determines the next value of the sequence counter based on the inputs from the PHY or on a PSP. There are 64 Branch Sequence Registers. Each register contains 21 bits. The

registers are written over the system bus and read by the Sequence Counter (part of the PHY Command Sequencer). The location that is accessed is determined by the value of iSeq from the Sequence Counter. There are two possible branch destinations in each sequence register with a separate set of branch conditions for each address. The branch conditions are evaluated in the Sequence Counter.

The Command and Control Sequence Registers (CCSeqR) contain the commands to be issued to the PHY on the next PSP as well as the time for the next PSP. There are ten possible commands contained in each CCSeqR. These commands are:

- SEQ_TxIdle
- SEQ_SendPRS
- SEQ_SendFrame
- SEQ_SendDel
- SEQ_RxIdle
- SEQ_PRSDet
- SEQ_PreambleDet
- SEQ_AmFiltPrld
- SEQ_ExtCmnd1
- SEQ_ExtCmnd2

The commands are sent to the PSP and Command Output Mux (in the PHY Command Sequencer) where they are multiplexed with commands from the SoftMAC Command Register (in the PHY Command Sequencer).

SEQ_ExtCmnd1 and SEQ_ExtCmnd2 are extra command lines for future expansion. They may be brought to an external pin for debugging purposes.

Each CCSeqR also contains bits to enable the PSP to the PHY (PSPOUT_EN) and to enable interrupts on PSP (PSPFIQ_EN, PSPIRQ_EN). SEQ_PSPCntrRst_En causes the PSP Counter in the PSP Generator (in the PHY Command Sequencer) to reset when a BRPrmbldet is received from the Input Control Enable Controls (in the PHY Command Sequencer).

PSP_TimeOffset contains the value of the PSP Counter in the PSP Generator when an internal PSP will be generated. An actual PSP to the PHY is generated in the Command Output Mux.

The CCSeqRs are written to over the system bus. The output of the CCSeqR's is accessed by the value of iSeq from the Sequence Counter (part of the PHY Command Sequencer).

The SoftMAC Command Registers allow the processor to issue commands directly to the PHY over the system bus, by-passing the PHY Command Sequencer.

The SoftMAC Command Registers include four registers:

- 1) RegSeqPSPCntr
- 2) RegSeqControl
- 3) RegSMControlPSP
- 4) RegSMControl

All the registers are set over the system bus when the System Bus Address Decoder (part of the PHY Command Sequencer) issues WSMCmndReg. ADDR[3:2] selects one of the four registers for write operations.

When RegSeqPSPCnt is written to, PSP_CNTEN is created. PSP_CNTEN passes to a Level Synchronizer where it is synchronized to the RxPHYClk to create PSP_CNTENSync. PSPCNTENSync is used by the PSP Generator (part of the PHY Command Sequencer) to enable the PSP Counter.

regSeqControl contains a bit that generates SMnSEQ. SMnSEQ is used by the PSP and command Output Mux (part of the PHY Command Sequencer) to determine the origin of commands and the PSP pulse to the PHY. When SMnSEQ is low, all commands to the PHY and the PSP are generated by the PHY Command Sequencer. When SMnSEQ is high, all commands to the PHY and the PSP are generated by the processor directly by the processor over the system Bus.

Writing to reqSMControl PSP will set SM_PSP high during the write cycle. SM_PSP is processed by a pulse Synchronizer to generate SM_PSPSync, which is one RxPHYClk pulse wide. SM_PSPSync is sent to the PSP and command Output Mux.

Writing to regSMControl sets the SoftMAC command lines. These lines correspond to the commands that may be issued in the Command and Control Sequence Register. The commands that may be issued are:

SM_TxIDLE

SM_SendPRS

SM_SendFrame

SM_SendDel

SM_PRSDet

Sm_PreambleDet

SM_RxIdle

Sm_AMFiltPrid

Sm_ExtCmnd1

SmExtCmnd2

These commands are then sent to the PSP and command Output Multiplexer blocks.

6. MAC/PHY Status and Interrupt Controller (#230)

The MAC/PHY Status Register and Interrupt Controller provides a single point of access to the status of the MAC/PHY and provides two interrupt signals from the MAC/PHY for use in a system interrupt controller. One interrupt, HMFIRQ, is intended to be used as a high priority interrupts at the system level. The second interrupt, HMIRQ, is intended to be used as a maskable interrupt at the system level.

7. PHY Register RD/WR Interface (#235)

The system bus interface provides address decode and read data select for HardMAC modules that are resident on the system bus. The system bus will provide a single system bus select line for the system MAC/PHY.

8. DCB CAM (#240)

The DCB-CAM (content addressable memory) accelerates the location of a Destination Control Block (DCB) based on the source address of an incoming HPA frame. When a source address (SA) is provided with SARdy, the DCB-CAM will return a pointer to the DCB associated with that source. If no match is found for the SA, the DCB-CAM will return a zero pointer.

9. Miscellaneous HardMAC Registers (#245)

The Miscellaneous HardMAC Registers contain simple registers and simple functions that do not belong in the other blocks of the HardMAC. There are three functions in the Miscellaneous HardMAC Registers. The FEC Uncorrectable Error Counter, the FEC Correctable Error Counter and the FCS Check Reset Register.

The FEC Uncorrectable Error Counter counts the number of uncorrectable FEC errors detected by the PHY while receiving a PLT frame.

The FEC Correctable Error Counter counts the number of FEC errors detected and corrected by the PHY while receiving a PLT frame.

The FCS Check Reset Register allows the SoftMAC to reset the FCS checker and all the associated registers.

CLAIMS

What is claimed is:

1. A very flexible MAC/PHY layer controller comprising programmable pre-defined operation hardware coprocessor modules including programmable parameterized functions, wherein the programmable coprocessor modules are coupled to a general purpose processor and hardwired DSP logic.
2. The controller of claim 1, wherein the hardware module can be easily adapted to changes in regulatory, device and end-product requirements with simple software changes.
3. The controller of claim 1, wherein the hardware modules are an implementation of a PLC MAC/PHY, targeted at an in-home environment.
4. The controller of claim 1, wherein the hardware modules are an implementation of a PLC MAC/PHY, targeted at an access environment.
5. The controller of claim 1, wherein the hardware modules are an implementation of a PLC MAC/PHY, targeted at an MDU/MTU environment.
6. The controller of claim 1, wherein the hardware modules are an implementation of a MAC/PHY targeted at any communications technology.

ABSTRACT OF THE DISCLOSURE

This invention defines a highly programmable MAC architecture for handling protocols that require precision timing and demand very short response times. The Media Access Controller consists of micro-coded programmable co-processors and general purpose CPUs. CPUs perform processing intensive functions while co-processors perform PHY specific media access control functions. The uniqueness of the architecture is in the real-time programmability of the co-processors; they can be reprogrammed by the CPUs based on the calculations performed in the CPU domain.

Any embodiment of this invention is suitable for ASIC, FPGA, discrete or combinations of these implementation schemes. The invention applies to any communications technology.

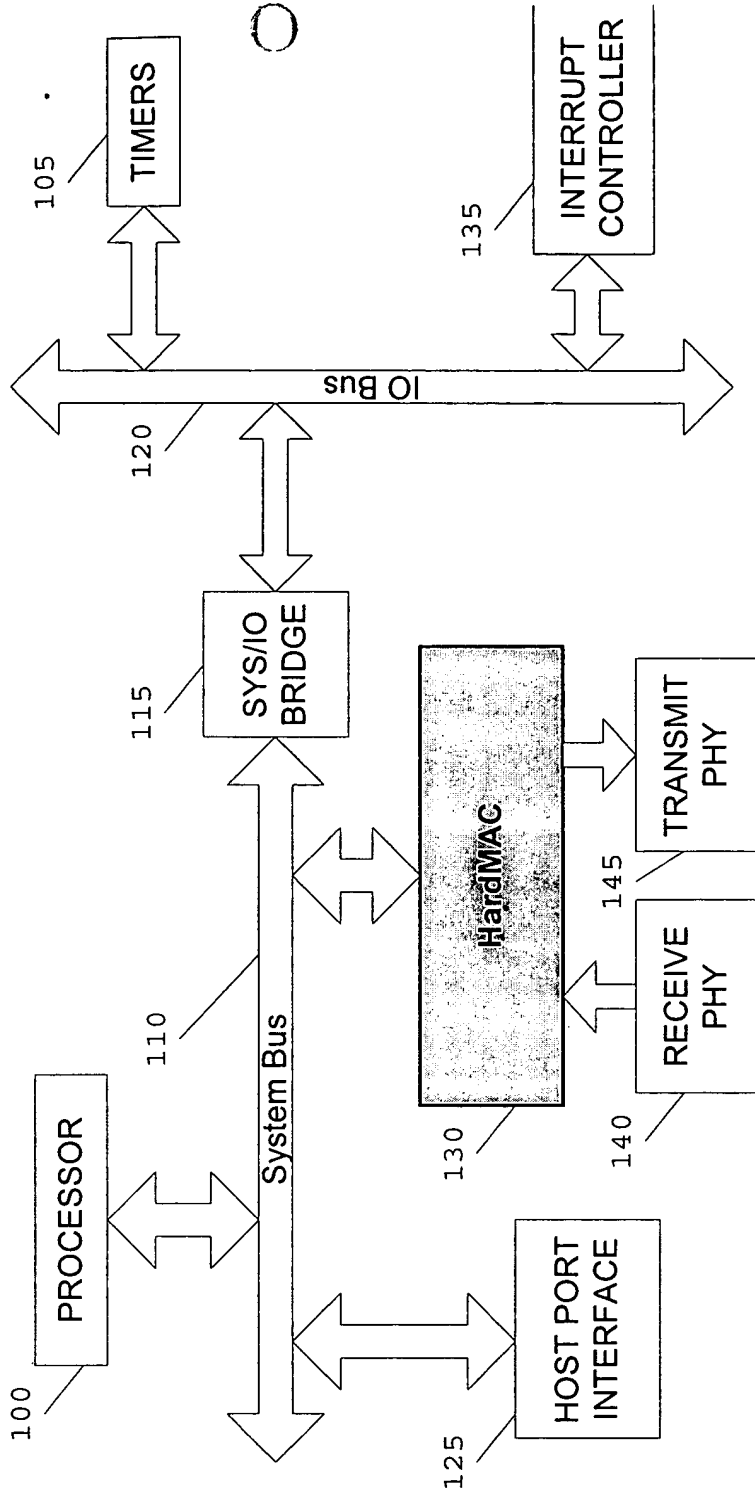


Figure 1: System Blocks

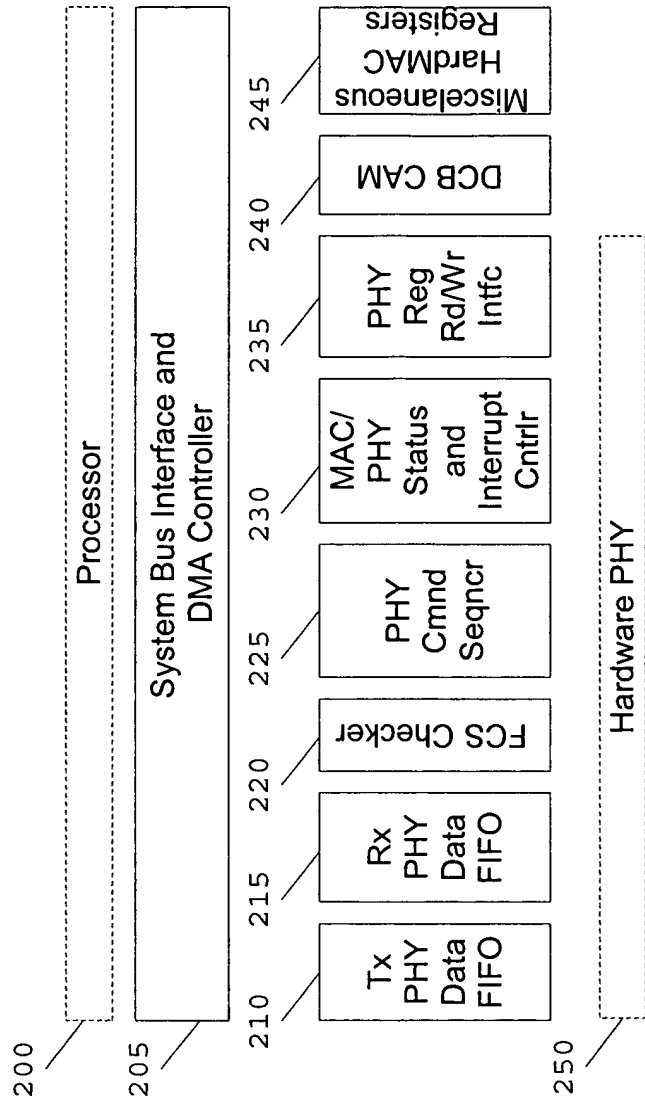


Figure 2: Major Internal Blocks

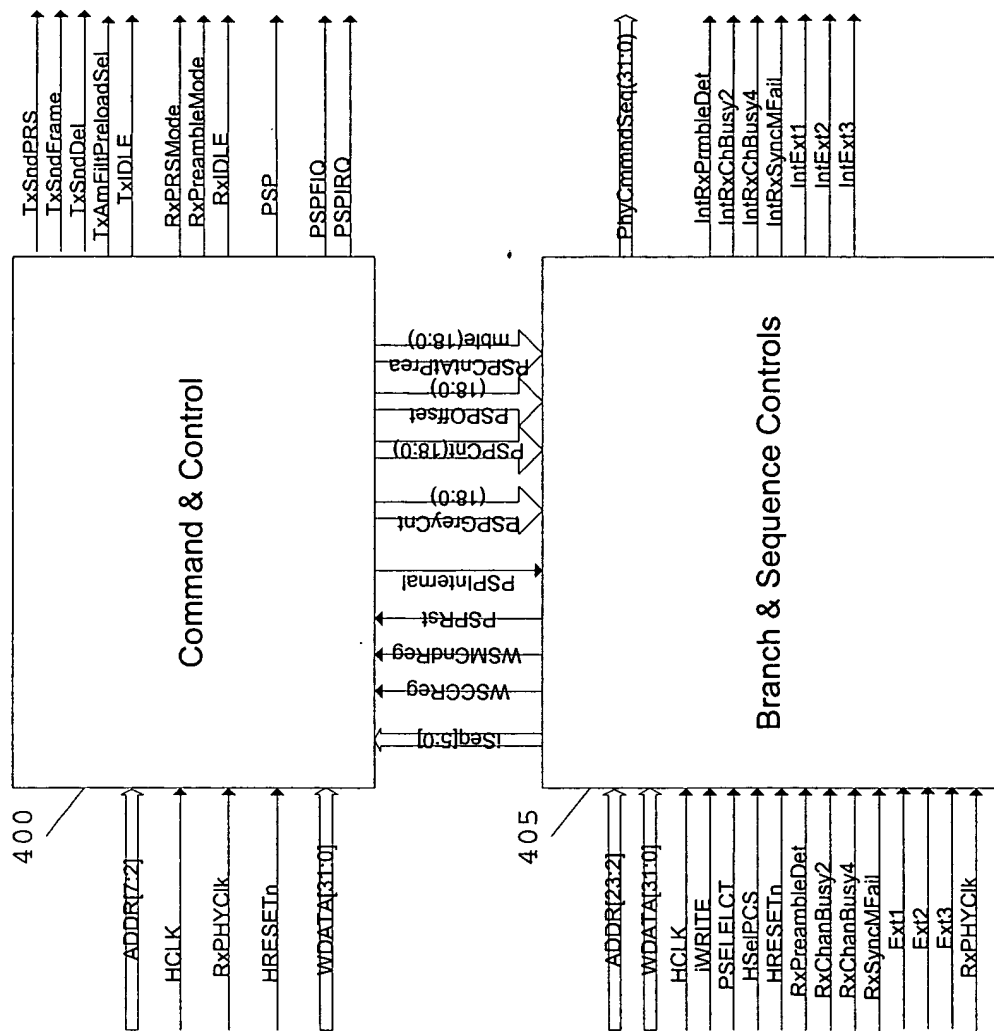


Figure 4: Example PHY Command Sequencer (#225)